Interpretable Survival Machine Learning for Predictive Maintenance

Lohithkumar J K, Prasad M N S, Sanjaykumar, Vishwas J H, Mayur N V
Assistant Professor, Dept. of ISE, APSCE, Bengaluru
Bengaluru, India
lohitjk@gmail.com

Abstract - **Predictive Maintenance aims to reduce downtime and costs by estimating when equipment is likely to fail. Traditional Remaining Useful Life models often lack interpretability and struggle with censored data. To address this, we propose a survival-based machine learning framework that integrates classical Cox Proportional Hazards models with ensemble learners like Random Survival Forests and Gradient Boosted Survival Machines. Instead of predicting a single failure time, this system estimates time-to-failure probabilities from multivariate sensor time series, accommodating both failed and censored equipment records. This probabilistic view enables more adaptive and risk-aware maintenance decisions. Compared to black-box RUL estimators, this approach offers greater transparency and reliability, making it better suited for industries that require interpretable and data-driven maintenance strategies.**

Keywords - Predictive Maintenance, Time-to-Failure Probabilities, Random Survival Forest

I. INTRODUCTION

Industrial machinery such as aircraft engines, gas turbines, power-generation units, and manufacturing equipment operates under dynamic and continuously varying conditions. Ensuring the reliability of these assets is critical for preventing catastrophic failures, reducing maintenance costs, and maximizing operational efficiency. Traditionally, industries relied on Scheduled Maintenance or Corrective Maintenance, where components are repaired or replaced either at fixed intervals or only after a failure has occurred. These practices often lead to unplanned downtime, excessive maintenance expenses, and safety risks. With the advancements of Industry 4.0 and the widespread adoption of interconnected sensors, Predictive Maintenance (PdM) has emerged as a data-driven solution that forecasts equipment degradation and predicts failures before they occur. Most modern PdM systems estimate Remaining Useful Life (RUL) using Machine Learning (ML) or Deep Learning (DL) models. Although these approaches often provide high prediction accuracy, they behave as black-box models, offering only a single numeric pre diction without explaining why a failure is expected. Additionally, these models struggle with censored data—instances where the machine has not failed during the observation period. They also do not quantify uncertainty, which is essential for realworld decision-making in safety-critical environments. To address these challenges, this work proposes an interpretable survival-based predictive maintenance framework that estimates time-to-failure probabilities instead of a single RUL prediction. Survival Analysis (SA)—a statistical technique widely used in medical prognosis and reliability engineering—provides tools such as hazard functions, survival curves, and risk scores, allowing engineers to understand both when a failure may occur and how risk evolves over time. Survival analysis also naturally incorporates censored data, making it highly suitable for industrial prognostics..

II. LITERATURE REVIEW

Syed Shazaib Shah et al. [1] propose a multi-parametric attention-based deep learning model—ForeNet-2D and ForeNet-3D—to forecast the remaining useful life (RUL) of wind turbines. These models incorporate temporal and spatial attention layers to accurately capture degradation trends in turbine systems. The results on large-scale wind turbine datasets show high short-term failure prediction accuracy. However, the reliance on domain-specific hyperparameters limits the generalizability of the approach to broader industrial contexts. Yicheng Zhang et al. [2] develop a CNN-based deep model for health index construction and RUL prediction using the NASA C-MAPSS dataset. Their framework fuses multi-sensor time-series data and applies temporal convolution to enhance feature extraction. The model achieves state-of-the-art performance on C-MAPSS; however, it lacks explainability, and its performance in transfer learning scenarios remains uncertain.

[3] Bin Liu et al. introduce a stacked autoencoder-based deep learning model for feature extraction and RUL prediction of industrial systems. The unsupervised pretraining followed by fine-tuning enables the model to learn compact representations from high-dimensional sensor data. Though the method reduces dependence on manual feature engineering, its performance deteriorates on small datasets, suggesting limited robustness.

Lisheng Tong et al. [4] present a hybrid LSTM-Autoencoder model for predictive maintenance of rotating machinery. The model first compresses high-dimensional input into a latent representation using an autoencoder and then leverages LSTM layers for temporal degradation learning. The hybrid approach performs better than individual models but is sensitive to noise in the input signals, revealing a gap in robustness under real-world noisy conditions. Juan Pablo García et al. [5] design a deep residual CNN architecture to predict equipment failure using high-frequency vibration and thermal data. Their model employs skip connections to mitigate vanishing gradients and successfully identifies early fault indicators. The model shows excellent generalization in lab settings; however, its performance in non-stationary or outdoor industrial environments is untested. designations.

[6] Seonghoon Woo et al. propose an attention-based Bi-LSTM network for RUL prediction of turbofan engines. The

attention mechanism improves interpretability by highlighting sensor readings that contribute most to the prediction. The model demonstrates superior accuracy compared to standard LSTM models but requires careful calibration of attention weights, which introduces complexity in deployment. [7] Yushi Tan et al. introduce a Temporal Convolutional Network (TCN) for remaining useful life estimation of mechanical systems. Their TCN model captures longrange dependencies more efficiently than recurrent networks. While the model exhibits promising results on the PHM08 dataset, its applicability to other domains with heterogeneous data remains unexplored.

[8] Xiaozhi Liu et al. present a dual-channel deep network architecture combining CNN and GRU layers for joint spatial-temporal modeling in predictive maintenance. The model is evaluated on electric motor datasets and achieves better failure prediction than baseline LSTM models. However, the authors highlight the challenge of interpretability in mixed-architecture models as a limitation. Songyi Zhang et al. [9] propose a Transformer-based model for RUL estimation of aircraft engines. The model uses self-attention to capture global dependencies in multivariate time-series data. It outperforms LSTM-based approaches but suffers from overfitting on smaller datasets, highlighting the need for regularization and data augmentation techniques.

[10] Yujie Yang et al. apply a GAN-based framework for predictive maintenance, where the generator learns to synthesize failure-like sensor readings while the discriminator predicts RUL. This adversarial training strategy enhances robustness to sensor noise. Nevertheless, the complexity of GAN training and mode collapse issues limit its practicality in resource-constrained environments. Christoph Helwig et al. [11] develop a machine learning pipeline using Random Forests and Support Vector Machines (SVM) to detect early degradation in rolling bearings. Their method emphasizes robust feature extraction from vibration signals using statistical and frequency-domain features. While the model performs well in lab experiments, the authors note difficulties in generalizing to real-world operating conditions without domain adaptation.

[12] Gang Niu et al. propose a semi-supervised predictive maintenance framework using label propagation and SVM. The system effectively leverages small labeled datasets to train accurate failure prediction models. However, the framework struggles when labeled data is highly imbalanced, suggesting the need for advanced resampling or cost-sensitive learning strategies. S. Widodo and B.-S. [13] Yang present a classification-based predictive maintenance system using decision trees and support vector regression for fault diagnosis of rotating machinery. The model demonstrates interpretability and ease of deployment. A major limitation is the requirement for wellstructured and noise-free input, which is rarely available in harsh industrial settings.

[14] Milos Jovanovic et al. design a real-time anomaly detection system using knearest neighbors (k-NN) and Random Forest classifiers to identify bearing faults. Their approach combines time-domain features with online learning for adaptive model updates. While effective in lab environments, scalability to large-scale IoT deployments is not addressed.

[15] Y. S. Abu-Mahfouz and M. A. Al-Ghandoor implement a predictive maintenance strategy using Naive Bayes and decision trees to identify failure trends in HVAC systems. Their comparative analysis shows that decision trees outperform probabilistic models in interpretability. However, the simplicity of the model architecture hinders its ability to capture nonlinear degradation patterns. [16] Mehmet Bektas et al. use Random Forest classifiers on the NASA C-MAPSS dataset to predict RUL in aircraft engines. They highlight the model’s robustness and ability to rank feature importance for diagnostics. A noted gap is the lack of timeseries modeling, which limits the framework’s ability to capture degradation trends over time.

Zhihua Chen et al. [17] evaluate Gradient Boosting Machines (GBMs) for early fault prediction in wind turbine gearboxes. Their method shows better performance compared to neural networks under small training datasets. However, GBMs face difficulties when dealing with sensor drift and multicollinearity in real-time applications. [18] Mahmoud Ahmad et al. present a hybrid feature selection and classification approach using ReliefF and ensemble learning for machine failure prediction. The study achieves high accuracy but lacks cross-domain validation, which questions the model’s generalizability across industries.

[19] Yujie Zhao et al. explore logistic regression combined with signal processing for binary classification of machine faults. Despite its simplicity, logistic regression achieves reasonable results on benchmark datasets. However, the method is unsuitable for capturing complex fault progression over time, limiting its role to binary anomaly detection. Fabien Campagne et al. [20] design a decision-tree-based framework for interpretable predictive maintenance in transportation fleets. Their model prioritizes transparency and rule-based logic to assist maintenance technicians. Though user-friendly, the model sacrifices predictive performance, especially when compared to deep learning alternatives.

[21] Salih K. et al. develop a hybrid predictive maintenance framework that integrates convolutional neural networks (CNNs) with random forests to diagnose rotating machinery faults. CNNs handle feature extraction from raw vibration signals, while random forests serve as the classifier. The combined model improves performance over standalone methods. However, real-time implementation remains a challenge due to computational load on embedded systems. [22] Yifan Li et al. present a hybrid prognostic model combining deep belief networks (DBNs) and autoregressive integrated moving average (ARIMA) models for remaining useful life (RUL) prediction. DBNs are used to capture nonlinear features, while ARIMA models provide temporal trends. Though accurate, the hybrid method requires complex tuning and suffers from interpretability issues.

[23] Muhammad Usama et al. propose an intelligent edge-cloud collaboration model for predictive maintenance using ensemble learning and LSTM. The edge device filters and pre-processes data, and the cloud handles model inference. This division optimizes latency and energy consumption, but raises concerns about model synchronization and data consistency across layers. [24] Felix Weber et al. introduce a real-time predictive maintenance architecture based on fog computing. It uses light-weight ML classifiers (like SVM and decision trees) at the edge and more advanced DL models in the fog layer. Their evaluation on an industrial conveyor belt system shows significant latency reductions. A noted limitation is that edge

devices require frequent retraining due to shifting machine conditions.

[25] Khaled ElFakharany et al. design a smart predictive maintenance solution using a fusion of LSTM and rule-based anomaly detection, deployed on industrial IoT platforms. Their dual-model design allows for both interpretability and deep learning accuracy. Yet, deployment is restricted to specific hardware setups, limiting scalability. [26] Antonio Angioni et al. implement a predictive maintenance system combining CNN-based feature learning and gradient boosted decision trees (GBDT) for classification. Tested on motor datasets, the hybrid system demonstrates enhanced robustness over traditional approaches.

Lohithkumar J K et al. [27] reviews the role of AI and Machine Learning in mechanical engineering, with special emphasis on predictive maintenance using deep learning and fault diagnosis techniques. It highlights how AI improves equipment reliability, reduces downtime, enhances manufacturing efficiency, and supports smart engineering applications while discussing challenges like data availability and model interpretability.

However, interpretability remains limited due to the black-box nature of CNNs.

III. MATERIALS AND METHODS

A. Dataset and Environment

This project used the NASA Commercial Modular Aero-Propulsion System Simulation (CMAPSS) turbofan engine dataset (commonly FD001 variant) as the experimental benchmark. The dataset contains multiple engine unit records. each recorded across operational cycles up to failure. Each row corresponds to one cycle of a particular engine unit and includes the following typical fields: Unit ID, Cycle, three operational settings (Setting1, Setting2, Setting3), and ~21 sensor measurements (Sensor1… Sensor21). For FD001, all units are run to failure (no censored records), which simplifies event labeling as in table 1. However, the pipeline supports censored records for other datasets.

TABLE 1.

CMAPSS Jet Engine Simulated Data		
Column Index	***Name***	***Description***
1	unit	Engine ID (1 to N engines)
2	time	Cycle number (starting from 1 for each engine)
3-5	op_setting_1 to op_setting_3	Operational settings (throttle, altitude, etc.)
6-26	sensor_1 to sensor_21	Sensor measurements (e.g., temperature, pressure, speed, vibration)

B. Data pre-processing

- Survival labels: create duration = cycles until failure (or censoring), and event indicator = 1 if failure observed, 0 if censored. For FD001 event = 1 for all final cycles.
- Avoid Missing / constant sensor handling: drop or impute sensors with excessive missing values or near-constant signals.
- Scaling: StandardScaler (zero mean, unit variance) for models that assume linear scale (Cox-PH, GBSM, DeepSurv). Tree-based models (RSF) are robust without scaling but consistent scaling was applied for reproducibility.
- Feature engineering: temporal features were added such as rolling means, short-term/long-term slopes, and last-k differences for selected sensors to capture degradation trends. These derived features often improve survival ranking performance.
- Train/validation/test split: split by unit (not by row) to avoid leakage: units were split into training (60–70%), validation (15–20%), and test (15–20%) sets.

C. Model Implementation

The project implements and compares four survival models: Cox Proportional Hazards (Cox-PH), Random Survival Forest (RSF), Gradient-Boosted Survival Machine (GBSM), and DeepSurv. Below are the implementation notes, library choices, and typical hyperparameter settings used for each model. Cox Proportional Hazards (Cox-PH): The Cox model was implemented using lifelines. CoxPH Fitter (or scikit-survival's Cox implementation). Continuous covariates were standardized before fitting. A small L2 penalizer (penalizer=0.01) was used to reduce variance due to correlated sensors. Diagnostics included checking proportional-hazards assumptions using Schoenfeld residuals.

Random Survival Forest (RSF): RSF was implemented using sksurv.ensemble.RandomSurvivalForest. RSF handles nonlinearity and interactions and provided the best performance in experiments. Key hyperparameters tuned via RandomizedSearchCV were n_estimators, min_samples_split, min_samples_leaf, and max_features. Example tuned configuration that produced best test C-Index in your run: n_estimators=300, min_samples_split=12, min_samples_leaf=13. OOB estimates were used for early validation where possible. Feature importance (permutation or impurity) and individual survival functions were extracted for interpretability.

Gradient-Boosted Survival Machine (GBSM): GBSM was used with sksurv.ensemble.GradientBoostingSurvivalAnalysis. The model was trained with a Cox partiallikelihood loss or a survival-specific objective where supported. Important hyperparameters: learning_rate, n_estimators, and max_depth. Typical settings: learning_rate=0.05, n_estimators=200–300, max_depth=3–5. SHAP analysis was used for model-agnostic interpretability of boosting outputs.

DeepSurv: DeepSurv (a neural network approximating the Cox risk function) was implemented using pycox + pytorch or a custom Keras/TensorFlow model. The network architecture was intentionally shallow to avoid overfitting given dataset size — e.g., two hidden layers with [128, 64] units, ReLU activations, dropout (0.2), and L2 weight decay. The Cox partial-likelihood loss was used, optimized with Adam and early stopping based on validation C-Index. GPU acceleration (Colab GPU) was enabled for faster training.

Hyperparameter Search & Reproducibility: Randomized search over the hyperparameter space was used to locate good configurations efficiently. Each model's best hyperparameters, validation scores, and training logs were saved. Random seeds were fixed for reproducibility and critical artifacts (best models, scaler objects) were saved to Google Drive

D. Evaluation Metrics

Model evaluation focused on ranking and probabilistic calibration metrics appropriate for censored survival data. The primary metric is the Concordance Index (C-Index); secondary metrics include the Integrated Brier Score (IBS) and time-dependent measures such as time-dependent AUC.

Concordance Index (C-Index): The Concordance Index (C-Index) is the primary evaluation metric used in survival analysis to measure how well a model can correctly rank individuals based on their predicted risk or time-to-failure. Instead of predicting an exact failure time, survival models output a risk score or survival probability, and the C-Index checks whether these predictions preserve the correct ordering of actual failure times. The C-Index considers pairs of data points and counts how many pairs are concordant, meaning the engine that actually fails earlier is also assigned a higher predicted risk (or shorter survival time) by the model. The score is computed as the ratio of correctly ordered pairs to the total number of comparable pairs.

A C-Index of 0.5 indicates random guessing, while a C-Index of 1.0 represents perfect ranking performance. Thus, a higher C-Index reflects a model's ability to correctly identify which engines are at higher risk of early failure. This makes the C-Index particularly important for predictive maintenance, where knowing which engine is more likely to fail sooner is more valuable than predicting the exact remaining cycles. In this project, the Random Survival Forest (RSF) achieved the highest C-Index score of 0.8543, showing strong ranking capability and confirming it as the most reliable model among those evaluated.

IV. RESULTS

This section presents the experimental outcomes obtained from implementing the proposed survival-based predictive maintenance framework on the NASA CMAPSS dataset (FD001). The goal of the analysis is to evaluate how effectively different survival models—Cox Proportional Hazards (Cox-PH), Random Survival Forests (RSF), Gradient-Boosted Survival Machines (GBSM), and DeepSurv—predict time-tofailure and rank engine units according to their failure risk. The performance evaluation primarily focuses on the Concordance Index (C-Index), survival curve behavior, feature importance, and interpretability outputs.

TABLE 2.

Results comparison	
Model	***C-Index Score***
Cox Proportional Hazards (Cox-PH)	0.811
Gradient-Boosted Survival Machine (GBSM)	0.8217
DeepSurv	0.8146
Random Survival Forest (RSF)	0.8543

Among all models, RSF achieved the highest C-Index, demonstrating superior ability to capture non-linear degradation patterns, sensor interactions, and complex survival relationships. Cox-PH performed reasonably well but was constrained by its proportional hazards assumption. DeepSurv improved non-linear modeling but required careful tuning to avoid overfitting. GBSM offered mid-level performance but was unable to surpass RSF.

V. CONCLUSION

This paper presented an interpretable, survival-based predictive maintenance framework designed to estimate time-to-failure probabilities for turbofan engines using the NASA CMAPSS dataset. Unlike traditional Remaining Useful Life (RUL) prediction models that provide only a single deterministic output, the proposed method leverages survival analysis to generate probabilistic insights, survival curves, and hazard functions, enabling richer and more transparent maintenance decision-making.

Four survival models—Cox Proportional Hazards (Cox-PH), Random Survival Forests (RSF), Gradient-Boosted Survival Machines (GBSM), and DeepSurv—were implemented, evaluated, and compared. Based on the Concordance Index (C-Index), RSF demonstrated the best performance with a score of 0.8543, outperforming both classical and deep learning–based survival models. The RSF model also offered meaningful interpretability through feature importance rankings, identifying key sensors related to temperature, pressure, vibration, and rotational speed as major contributors to engine degradation.

The survival curves generated for selected engine units further highlighted the advantages of the proposed approach. Critical maintenance thresholds were identified, such as 99% survival probability at 57 cycles, 90% survival probability at 95 cycles, and 50% probability at 157 cycles, providing actionable insights for preventive maintenance scheduling. Additionally, the framework addressed limitations of existing RUL-based methods by incorporating censored data handling, risk estimation, and interpretable model outputs.

Overall, the results indicate that survival analysis offers a robust, transparent, and practically deployable alternative to conventional black-box RUL prediction systems. The integration of survival models into predictive maintenance pipelines can significantly enhance reliability, optimize maintenance planning, and support risk-aware decision-making across industrial environments

REFERENCES

[1] S. S. Shah, D. Tan, and C. K. Sah, "RUL forecasting for wind turbine predictive maintenance based on deep learning," Heliyon, vol. 10, p. e39268, Oct. 2024.

[2] Y. Zhang, X. Qin, and L. Liu, "A CNN-based deep learning model for remaining useful life prediction using multi-sensor time series," Sensors, vol. 22, no. 1, pp. 1–14, 2022.

[3] B. Liu, Z. Wang, and Y. Yang, "Stacked autoencoder-based deep learning for RUL prediction of industrial equipment," Computers in Industry, vol. 98, pp. 192–202, 2018.

[4] L. Tong, H. Liu, and W. Zhou, "A hybrid LSTM-autoencoder model for predictive maintenance of rotating machinery," Journal of Manufacturing Systems, vol. 63, pp. 91–102, 2022.

[5] J. P. García, F. Rivas, and A. Ortega, "Residual CNN-based predictive maintenance for rotating machines," IEEE Transactions on Industrial Informatics, vol. 17, no. 5, pp. 3317–3325, 2021.

[6] S. Woo, M. Kang, and J. Kim, “Attention-based Bi-LSTM model for remaining useful life prediction,” IEEE Access, vol. 9, pp. 107222–107231, 2021.

[7] Y. Tan, J. Wu, and H. Guo, “Remaining useful life prediction using temporal convolutional networks,” Reliability Engineering & System Safety, vol. 215, p. 107868, 2022.

[8] X. Liu, J. Luo, and K. Chen, “Dual-channel deep learning model for predictive maintenance of electric motors,” IEEE Transactions on Industrial Electronics, vol. 69, no. 4, pp.3202–3211, 2022.

[9] S. Zhang, Y. Huang, and G. Lin, “Transformer-based remaining useful life prediction for aircraft engines,” IEEE Access, vol. 10, pp. 54821–54831, 2022.

[10] Y. Yang, Z. Sun, and B. Liu, “GAN-based adversarial training for robust RUL estimation,” Expert Systems with Applications, vol. 185, p. 115645, 2021.

[11] C. Helwig, H. Schaefer, and W. Sextro, “Condition monitoring of rolling bearings using machine learning,” Mechanical Systems and Signal Processing, vol. 88, pp. 100–112, 2017.

[12] G. Niu, B. Dai, and S. Wang, “A semi-supervised approach for predictive maintenance with imbalanced labels,” IEEE Transactions on Industrial Informatics, vol. 17, no. 4, pp. 2658–2666, Apr. 2021.

[13] S. Widodo and B.-S. Yang, “Support vector machine in machine condition monitoring and fault diagnosis,” Mechanical Systems and Signal Processing, vol. 21, no. 6, pp. 2560– 2574, 2007.

[14] M. Jovanovic, T. Andelkovic, and V. Milinkovic, “A k-NN and Random Forest approach for real-time fault detection in bearings,” Expert Systems with Applications, vol. 130, pp. 145–155, 2019.

[15] Y. S. Abu-Mahfouz and M. A. Al-Ghandoor, “A decision-tree-based approach for HVAC failure prediction,” Energy and Buildings, vol. 159, pp. 1–9, 2018.

[16] M. Bektas, O. Fidan, and E. Aslan, “Remaining useful life estimation using random forests on C-MAPSS dataset,” Procedia Computer Science, vol. 168, pp. 145–150, 2020.

[17] Z. Chen, H. Yang, and Y. Liu, “Fault prediction in wind turbine gearboxes using gradient boosting,” Renewable Energy, vol. 145, pp. 2059–2069, 2020.

[18] M. Ahmad, S. Imran, and A. Iqbal, “Predictive maintenance using feature selection and ensemble learning,” Applied Soft Computing, vol. 103, p. 107142, 2021.

[19] Y. Zhao, X. Li, and M. Ding, “Binary classification of machine faults using logistic regression and time-domain features,” Measurement, vol. 143, pp. 1–9, 2019.

[20] F. Campagne, J. Dubois, and C. Moreau, “Interpretable predictive maintenance for transport fleets using decision trees,” Transportation Research Part C: Emerging Technologies, vol. 98, pp. 72–83, 2019.

[21] S. K. Yildirim and M. C. Acarman, “A hybrid approach to fault diagnosis using CNN and random forest in rotating machinery,” Measurement, vol. 173, p. 108578, 2021.

[22] Y. Li, X. Zhou, and J. Wu, “Remaining useful life prediction using deep belief networks and ARIMA,” Journal of Intelligent Manufacturing, vol. 32, pp. 395–408, 2021.

[23] M. Usama, Y. Ben-Hur, and A. Shahid, “Edge-cloud collaborative learning for predictive maintenance using ensemble and LSTM,” Future Generation Computer Systems, vol. 129, pp. 246–259, 2022.

[24] F. Weber, T. Fischer, and H. Hinkenjann, “Fog-based real-time predictive maintenance using machine learning,” Procedia Computer Science, vol. 184, pp. 220–229, 2021.

[25] K. ElFakharany, M. Abdelhameed, and M. S. Elmogy, “IoT-enabled hybrid predictive maintenance using LSTM and rule-based logic,” Computers in Industry, vol. 134, p. 103563, 2022.

[26] A. Angioni, M. Carta, and G. Fenu, “Combining CNN and GBDT for effective predictive maintenance,” IEEE Access, vol. 9, pp. 87654–87666, 2021.

[27] Lohithkumar J. K., Deepak A. R., Rakshithkumar P., and Roopa D., “AI and Machine Learning in Mechanical Engineering: Innovations and Future Prospects—Review,” International Journal of Engineering Research & Technology (IJERT), vol. 13, no. 3, 2025.